



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/540,576	03/31/2000	Robert G. Field	SUNIP252/P4198	2536

22434 7590 09/24/2003

BEYER WEAVER & THOMAS LLP
P.O. BOX 778
BERKELEY, CA 94704-0778

[REDACTED] EXAMINER

KISS, ERIC B

ART UNIT	PAPER NUMBER
----------	--------------

2122

DATE MAILED: 09/24/2003

17

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/540,576	FIELD ET AL.	
	Examiner	Art Unit	
	Eric B. Kiss	2122	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on 10 July 2003.
 2a) This action is FINAL. 2b) This action is non-final.
 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1-32 is/are pending in the application.
 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
 5) Claim(s) _____ is/are allowed.
 6) Claim(s) 1-32 is/are rejected.
 7) Claim(s) _____ is/are objected to.
 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
 10) The drawing(s) filed on 30 October 2002 is/are: a) accepted or b) objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 11) The proposed drawing correction filed on _____ is: a) approved b) disapproved by the Examiner.
 If approved, corrected drawings are required in reply to this Office action.
 12) The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

- 13) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
 * See the attached detailed Office action for a list of the certified copies not received.
 14) Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).
 a) The translation of the foreign language provisional application has been received.
 15) Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) Paper No(s). _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. The amendment of July 10, 2003, has been received and entered. Claims 1-32 are pending.

Response to Arguments

2. Applicant's arguments filed July 10, 2003, have been fully considered but they are not persuasive.

a) In response to Applicant's arguments on page 8, in paragraph 3, the client debugger object produces a connection object as asserted in the office action as the front-end debugger portion. The second citation of the client debugger object appears to have caused confusion because of its placement in the middle of the phrase "front-end debugger portion". However, the client debugger object reference is intended only to indicate an interpretation of the adjective phrase "front-end debugger", and it is the parenthetical reference following the word "portion", namely the connection object, that is intended to be an interpretation of the actual "front-end debugger portion produced. The rejection is not intended to assert that the client debugger object generates itself, but rather generates a connection object, as closer inspection of the rejection reveals.

Art Unit: 2122

b) As per Applicant's arguments on page 8, in paragraph 4, the following responses to previous arguments, as presented in the previous office action (Paper No. 15 mailed April 10, 2003), are reproduced below:

Applicant acknowledges that the "TPrimitiveConnection" object disclosed by You et al. defines a protocol for communication between a client and a server (see the last sentence of paragraph 3 on page 4 of Paper No. 14 -- Applicant's response filed March 26, 2003). In addition, it has been submitted that an accepted definition of the term "specification" is: A detailed description of something (see p.325 of Microsoft Press Computer User's Dictionary, 1998). It has been further submitted that the "TPrimitiveConnection" provides a formal detailed description of the communication protocol between a client and a server. The TPrimitiveConnection class disclosed by You et al. is used within C++ source code to instantiate objects or define subclasses from which objects are instantiated. C++ is a well-known object-oriented programming language, which is implemented using a compiler comprising a code generator to process the source code to produce machine-executable computer program code. Therefore, the Examiner has maintained that the TPrimitiveConnection disclosed by You et al. can be considered a formal specification which can be put into a code generator.

Furthermore, You et al. disclose TPrimitiveConnection as C++ source code defining an abstract base class (see column 52, line 30 through column 53, line 5). You et al. further disclose, "Communication between client and server are handled using TPrimitiveConnection objects" (see column 52, lines 8 and 9). This implies that the TPrimitiveConnection base class is used to generate TPrimitiveConnection objects. These objects are instances of the TPrimitiveConnection class or of a TPrimitiveConnection subclass (see column 52, lines 20-23).

Art Unit: 2122

In either case, this inherently requires the parsing the TPrimitiveConnection class definition using a code generator in order to generate the corresponding code for the object instances of the class.

You et al. disclose that the TPrimitiveConnection base class is used to generate TPrimitiveConnection objects, as described above. These connection objects in turn provide a front-end debugger (client) portion and a back-end debugger (server) portion (see column 52, lines 8-19). You et al. further disclose the front-end debugger program (client debugger) and back-end debugger program (debugger server) being compatible with each other (see Fig. 2, in which the client debugger is shown as interfacing with the debugger server).

In light of the above arguments, the Examiner has maintained that You et al. disclose inputting a formal specification into a code generator, which in turn parses the formal specification to generate a front-end debugger portion and back-end debugger portion, such that the front-end debugger program and the back-end debugger program are compatible with each other. Moreover, the Examiner maintains that “TPrimitiveConnection”, as disclosed by You et al. provides the formal specification. See the 35 USC § 102 and 35 USC § 103 rejections that follow.

- c) In response to Applicant's arguments regarding the rejections based on 35 U.S.C. §112, second paragraph, the Examiner submits that the trademark JAVA is owned by the present assignee of the instant application. As such, the present assignee is the sole producer and/or licensor of JAVA products and has the ability to modify the technology identified under the trademark JAVA. Thus, the trademark JAVA identifies the source of the products and not the

Art Unit: 2122

products themselves. In contrast, for example, C++ is a name used in trade to identify a particular nonproprietary programming language conforming to an accepted standard. Products and services incorporating the name C++ are produced by numerous sources. Further, the technologies identified using the trademark JAVA are continuously evolving. An example of this evolution can be found in “JSR 14: Add Generic Types To The Java™ Programming Language”, which describes a proposed amendment to the JAVA Language Specification submitted by Sun Microsystems, Inc., in 1999 and pending approval by the JAVA COMMUNITY PROCESS Program. In view of the statements presented above, it is asserted that the trademark JAVA has no fixed definite technical meaning. Accordingly, a rejection under 35 U.S.C. 112, second paragraph, based on the use of the trademark JAVA as a limitation in a claim, is proper.

d) On page 10, under the heading “Rejections based on Presentations of Java one Conference”, Applicant makes the following statement:

It is earnestly believed that the Examiner has made reference to material presented in Java two conference. Clarification is required.

However, the Examiner is at a loss in interpreting Applicant’s statement in any meaningful manner. Applicant has not pointed out a specific instance of a reference to a “Java two conference” in any previous office action, and the Examiner has only relied upon presentation slides from the JavaOne® 1998 conference as supplied by Applicant in making the rejections under 35 U.S.C. §§102(b), 103(a). In fact, the Examiner is not even aware of the existence, past or present, of any “Java two conference”.

e) In response to Applicant's challenge to the Official Notices taken in the previous office action, the Examiner respectfully submits that in each case, evidence supporting the Examiner's statements was either supplied in the previous office action or is being supplied herewith as summarized below.

On page 12, paragraph 1, of Paper No. 15 (the office action mailed April 10, 2003), the Examiner made the following statements using Official Notice:

Official Notice is taken that in order to arrive at a machine-readable implementation of a human-readable specification, a compilation process comprising parsing the input specification and generating the output code has been well-known and commonly practiced in the computer art. An exemplary description of this practice can be found in Alfred V. Aho, et al., "Compilers, Principles, Techniques, and Tools," 1986, Addison-Wesley (hereinafter *Aho et al.*). For instance, Fig. 1.9 on page 10 of *Aho et al.* shows the phases of such a compilation process, including syntax analysis (or parsing) and code generation. The alternative to compiling is writing the code directly in assembly language or binary machine language, which is typically impractical.

In the above-cited statements, evidence has already been provided in support of the Examiner's position. The act of compiling source code is exceedingly well known in the computer arts, as are the basic components, including parsers and code generators, that typical compilers comprise. The cited text of *Aho et al.* provides an introductory teaching of compilers and their components and as such, provides a description of a compiler, and proper motivation to use a compiler in translating computer code.

Art Unit: 2122

In the last paragraph of page 12, continuing onto page 13, of Paper No. 15, the Examiner made the following statements using Official Notice:

Official Notice is taken that the specific procedures and data packet formats necessary for sending and receiving data for a particular protocol are necessary in order to be able to implement such. One would be motivated to generate documentation of a communication protocol to provide human-readable protocol documentation information to software developers enabling them to implement the protocol. Further, HTML is a platform-independent document format, and one would be motivated to use HTML for the purpose of generating the documentation to allow it to be read on different platforms.

The above-cited Official Notice is based upon the assertions made by the Examiner, and unchallenged by the Applicant, in Paper Nos. 4 and 9 (the office actions mailed July 17, 2002 and November 20, 2002, respectively). In Paper No. 4 (see the rejection of claims 7, 11, and 16 under 35 U.S.C. §103(a) as unpatentable over U.S. Patent No. 5,787,245 to *You* on pages 10-11), the Examiner asserted the following:

[O]ne having ordinary skill in the computer art would recognize that the specific procedures and data packet formats necessary for sending and receiving data for a particular protocol are necessary in order to be able to implement such. One would be motivated to generate documentation of a communication protocol to provide human-readable protocol documentation information to software developers enabling them to implement the protocol. Further, HTML is a platform-independent document format, and one would be motivated to use HTML for the purpose of generating the documentation to allow it to be read on different platforms.

In Paper No. 9 (see the rejection of claims 7, 11, and 16 under 35 U.S.C. §103(a) as unpatentable over U.S. Patent No. 5,787,245 to *You* on pages 11-12), the Examiner maintained and reproduced the same unchallenged assertion for Applicant's consideration. Applicant's responses to Papers Nos. 4 and 9 (namely Paper Nos. 8 and 11, respectively) failed to seasonably challenge the above-cited Examiner's assertion.

Art Unit: 2122

If applicant does not seasonably traverse the well known statement during examination, then the object of the well known statement is taken to be admitted prior art. *In re Chevenard*, 139 F.2d 71, 60 USPQ 239 (CCPA 1943). A seasonable challenge constitutes a demand for evidence made as soon as practicable during prosecution. Thus, applicant is charged with rebutting the well known statement in the next reply after the Office action in which the well known statement was made. This is necessary because the examiner must be given the opportunity to provide evidence in the next Office action or explain why no evidence is required. If the examiner adds a reference to the rejection in the next action after applicant's rebuttal, the newly cited reference, if it is added merely as evidence of the prior well known statement, does not result in a new issue and thus the action can potentially be made final.

Further evidence in support of the above Official Notice is submitted herewith. "HTTP-NG Binary Wire Protocol," W3C Working Draft, July 10, 1998, is an example of a human-readable protocol description in HTML providing necessary information to implement the HTTP-NG Binary Wire Protocol. This document specifies, among other things, the message and data formats that the protocol supports or requires. Without this information, one of ordinary skill in the art would not be able to design a system or software utilizing this protocol.

Applicant's responses throughout the prosecution history have not properly addressed the basis for the rejection under 35 U.S.C. §102(b), based on public use, namely the exact nature and extent of the Inventors' disclosure related to the presentation titled "The New Java™ Platform Debugger Architecture," given in a public forum on March 26, 1998. It is further noted that in

Art Unit: 2122

Paper No. 12 (the Advisory Action mailed February 11, 2003), the Examiner laid out the steps considered necessary to overcome the aforementioned rejection as follows (emphasis added):

The rejection of claims 1-18 under 35 U.S.C. §102(b) is based on a presentation given by the Applicant. As such, the Applicant is in the best position to distinguish the material actually presented (in addition to information disclosed on the slides) from that which is claimed. The Applicant has stated that the presentation did not teach or suggest many of the recited features of the claimed invention. However, it is unclear to the Examiner as to whether this statement is made in regards to the entire presentation, including any oratory disclosure, or to only the documents of record. **Accordingly, the rejection is maintained, but will be withdrawn if the Applicant makes an affirmative statement on the record regarding claimed limitations not disclosed by the Applicant in any component of the March 26, 1998, presentation.**

The present challenges to the Official Notices taken (which have been addressed, along with supporting evidence believed to be adequate, as set forth above) appear to be, directly or indirectly, an attempt to delay prosecution of the instant application rather than an effort, in good faith, to resolve the outstanding issues raised by the evidence of public use presented by the Examiner. Again, if convincing evidence can be brought forward showing that the above-cited presentation did not, in fact, disclose the same invention which is presently claimed nor was the presently claimed invention embodied in any product made available or otherwise demonstrated or discussed prior to one year before the instant application's filing date, then the Examiner will rest his case and withdraw the rejections under 35 U.S.C. §102(b), based on public use of the invention.

Claim Rejections - 35 USC § 112

3. The following is a quotation of the second paragraph of 35 U.S.C. 112:

Art Unit: 2122

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

4. Claims 6, 7, 10, 13, 14, 17, 19, and 21- 23 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claims 6, 7, 10, 13, 14, 17, and 19 contain the trademark/trade name JAVA. Where a trademark or trade name is used in a claim as a limitation to identify or describe a particular material or product, the claim does not comply with the requirements of 35 U.S.C. 112, second paragraph. See *Ex parte Simpson*, 218 USPQ 1020 (Bd. App. 1982). The claim scope is uncertain since the trademark or trade name cannot be used properly to identify any particular material or product. A trademark or trade name is used to identify a source of goods, and not the goods themselves. Thus, a trademark or trade name does not identify or describe the goods associated with the trademark or trade name. In the present case, the trademark JAVA is improperly relied upon in the claim to incorporate the technical features of a particular programming language environment. However, the trademark JAVA can only properly define the source of the programming language environment, namely Sun Microsystems, Inc. Accordingly, the identification/description is indefinite.

Claims 21 and 22 each contain a limitation beginning with the phrase “can be”. These limitations are not positively recited, making it unclear whether recited features following the phrase are part of the claimed invention.

Claim 22 recites the limitation “wherein the transport mechanism can be a socket, or a serial line, or socket, or a shared memory implementation”. In this limitation, “socket” is listed twice.

Claim 23 recites a functional negative step (lines 6-8) that does not appear to be a natural result of previously recited method steps.

Claim Rejections - 35 USC § 102

5. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

6. Claims 1, 8, 12, 15, and 18 are rejected under 35 U.S.C. 102(b) as being anticipated by U.S. Patent No. 5,787,245 to You et al.

You et al. disclose inputting a formal specification (TPrimitiveConnection; see column 52, lines 8-27) into a code generator (client debugger object) which in turn parses the formal specification to generate a front-end debugger (client debugger object; see column 4, lines 28-37) portion (connection object; see column 63, lines 25-28) and a back-end debugger (server debugger object) portion (reverse connection object; see column 57, lines 35-41) based on the parsing of the specification. A communication protocol is enabled between the front-end debugger (client debugger object) and the back-end debugger program (server debugger object), wherein the communication protocol is defined by the formal specification

Art Unit: 2122

(TPrimitiveConnection). You et al. further disclose a computer readable medium including computer program code (see column 80, lines 33-65) and a computer system (see column 79, lines 13-55) for performing the aforementioned actions. You et al. further disclose the front-end debugger program (client debugger) and back-end debugger program (debugger server) being compatible with each other (see Fig. 2, in which the client debugger is shown as interfacing with the debugger server), and the components inherently comply with the specification from which they are generated.

Claim Rejections - 35 USC § 103

7. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

8. Claims 2, 3, 9 and 13 are rejected under 35 U.S.C. 103(a) as being unpatentable over You et al. as applied to claims 1 and 12, respectively, above.

As per claims 2 and 3, although You et al. disclose with such a C++ object-oriented programming language implementation and fails to disclose a Java object-oriented programming language method, one having ordinary skill in the computer art would recognize that the You et al. system can be implemented using a wide number of known object-oriented programming languages, including the Java programming language. Therefore, it would have been obvious to one having ordinary skill in the computer art at the time the invention was made to utilize Java programming language code running on a virtual machine to implement the method of You et al.

One would be motivated to do so in order to gain the platform independence that the Java programming language provides.

As per claim 9, although You et al. fail to teach the use of a declarative language, one having ordinary skill in the computer art would recognize that a specification could be written in any programming language style, including such a known declarative language. One would be motivated to do so because a declarative language is rule-based and is best suited to implementing a specification that is also rule-based. Therefore, it would have been obvious to one having ordinary skill in the computer art at the time the invention was made to write the formal specification of You et al. in a declarative language because it is best-suited for such a purpose.

As per claim 13, although You et al. disclose with such a C++ object-oriented programming language implementation and fails to disclose a Java object-oriented programming language method, one having ordinary skill in the computer art would recognize that the You et al. system can be implemented using a wide number of known object-oriented programming languages, including the Java programming language. One would be motivated to do so in order to gain the platform independence that the Java programming language provides. Further, although You et al. fail to teach the use of a declarative specification language, one having ordinary skill in the computer art would recognize that a specification could be written in any programming language style, including such a known declarative language. One would be motivated to do so because a declarative language is rule-based and is best suited to implementing a specification that is also rule-based. Therefore, it would have been obvious to one having ordinary skill in the computer art at the time the invention was made to write the

formal specification of You et al. in a declarative language because it is best-suited for such a purpose and to utilize Java programming language code running on a virtual machine to implement the front-end of the You et al. method to gain platform independence.

9. Claims 4 and 5 are rejected under 35 U.S.C. 103(a) as being unpatentable over You et al. as applied to claim 1 above, and further in view of U.S. Patent No. 5,901,315 to Edwards.

You et al. fail to teach the back-end debugger program, a portion of which comprising C language code, directly controlling and communicating with a virtual machine. However, Edwards teaches a back-end debugger program (debug engine, DE, and BE) comprising C language code (see column 4, lines 35-38) that directly controls and communicates with a virtual machine (see Figure 3). One having ordinary skill in the computer art would recognize that a back-end debugger program could be written in any known programming language that allows an interface to be established between a debuggee program and a debugger front-end. Further, a virtual machine that is controlled by and communicates with the debugger back-end is commonly used when the application being debugged comprises Java language code. It would have been obvious to one having ordinary skill in the computer art at the time the invention was made to implement the teachings of Edwards into the method of You et al. in order to get the advantage of being able to interface with and debug a Java language program. One would be motivated to do so for debugging an application comprising Java language code using a non-Java language user interface.

Art Unit: 2122

10. Claims 6, 10, and 14 are rejected under 35 U.S.C. 103(a) as being unpatentable over You as applied to claims 1, 8, and 13 above, and further in view of Field et al., "The New Java Platform Debugger Architecture," contained in Birds of a Feather, '98 JavaOne conference schedule (hereinafter Field et al.).

Although You et al. disclose with such a protocol defined by a TPrimitiveConnection class, one having ordinary skill in the computer art would recognize that any known communication protocol could be used to implement the You et al. method and system, including a Java Debug Wire Protocol as once taught by Field et al. as a communication protocol between a debugger and a debuggee. One would be motivated to use the Java Debug Wire Protocol because it allows for cross-platform remote debugging. Therefore, it would have been obvious to one having ordinary skill in the computer art at the time the invention was made to incorporate the Java Debug Wire Protocol into the method and system of You et al. to perform cross-platform debugging.

11. Claims 7, 11, 16, and 17 are rejected under 35 U.S.C. 103(a) as being unpatentable over You et al. as applied to claims 6, 8, and 15 above.

As per claims 7, 11, and 16, although You et al. do not disclose a method of, or computer code for, generating HTML documentation of the protocol, one having ordinary skill in the computer art would recognize that the specific procedures and data packet formats necessary for sending and receiving data for a particular protocol are necessary in order to be able to implement such. One would be motivated to generate documentation of a communication protocol to provide human-readable protocol documentation information to software developers

Art Unit: 2122

enabling them to implement the protocol. Further, HTML is a platform-independent document format, and one would be motivated to use HTML for the purpose of generating the documentation to allow it to be read on different platforms. Therefore, it would have been obvious to one having ordinary skill in the computer art at the time the invention was made to incorporate the generation of HTML protocol documentation into the method and computer code of You et al. to allow software developers using various computer platforms to read and understand the proper procedures involved in implementing the protocol.

As per claim 17, see rationale provided in item 14 above.

12. Claims 1-32 are rejected under 35 U.S.C. 102(b) based upon a public use or sale of the invention or, in the alternative, under 35 U.S.C. 103(a) as obvious over the presentation given in a public forum on March 26, 1998 by the Applicant as evidenced by “JavaOne 1998 Presentation” (submitted in Information Disclosure Statement filed October 30, 2002 and hereinafter *Slideshow*), along with “Birds of a Feather, '98 JavaOne conference schedule”, (cited in previous office action and hereinafter *Schedule*).

The cited presentation (given in a public forum on March 26, 1998; see Schedule, p. 34) disclosed:

a formal specification defining a communication protocol written in JAVA Debug Wire Protocol (JDWP) declarative specification language (see, for example, pages 1, 2, 4, and 13 of *Slideshow*);

a JAVA front-end debugger program portion running on a first virtual machine (see, for example, pages 2-4, 7, 8, and 10-14 of *Slideshow*);

the JAVA front-end debugger program portion comprising JAVA programming language code (see, for example, pages 2-4, 7, 8, and 10-14 of *Slideshow*);
a back-end JAVA debugger program portion controlling and communicating with a second virtual machine (see, for example, pages 2, 4-8, and 10-14 of *Slideshow*);
the back-end debugger program portion comprising C language code (see, for example, pages 4 and 5 of *Slideshow*);
the JAVA front-end debugger program and JAVA back-end debugger program being compatible with each other (see, for example, pages 2, 4, 7, 8, and 10-14 of *Slideshow*);
a transport mechanism such as serial or socket between the front-end and back-end (see, for example, pages 2, 4, 7, 8, and 10-14 of *Slideshow*);
the formal specification being independent of (and not defining the particulars of) the transport mechanism (see, for example, page 4 of *Slideshow*);
sending events generated in the second virtual machine to the front-end via the back-end debugger program code portion (see, for example, pages 3-6 and 13-19 of *Slideshow*);
the front-end reading and parsing events from the back-end debugger code portion (see, for example, pages 3-6 and 13-19 of *Slideshow*);
the front-end processing module performing operations related to requests made through the front-end debugger program by the debugger application program (see, for example, pages 3-6 and 13-19 of *Slideshow*);

the front-end processing module writing formatted requests (see, for example, pages 3-6 and 13-19 of *Slideshow*);

the back-end processing module performing operations related to event processing and request processing (see, for example, pages 3-6 and 13-19 of *Slideshow*);

the event processing operations including sending an event which was generated through the virtual machine debugging interface to the front-end debugging portion (see, for example, pages 3-6 and 13-19 of *Slideshow*);

the request processing operations including reading and parsing formatted requests from the front-end debugger program portion (see, for example, pages 3-6 and 13-19 of *Slideshow*); and

the front-end debugger program portion including a class which is used by the front-end debugger program portion to send and receive information over the debugging communication protocol (see, for example, pages 3-6 and 13-19 of *Slideshow*).

Furthermore, the computer-readable medium and system of claims 15 and 18, respectively, are considered inherent in implementing the disclosed features described above.

It is unclear, based on materials made available to the Examiner, whether or not the cited presentation expressly disclosed inputting the formal specification into a code generator, parsing the formal specification, and generating the JAVA front-end debugger program portion and back-end JAVA debugger program portion from the formal specification after parsing. However, the front-end and back-end debugger program portions are disclosed as based on an implementation of the JDWP specification. Official Notice is taken that in order to arrive at a

machine-readable implementation of a human-readable specification, a compilation process comprising parsing the input specification and generating the output code has been well-known and commonly practiced in the computer art. An exemplary description of this practice can be found in Alfred V. Aho, et al., "Compilers, Principles, Techniques, and Tools," 1986, Addison-Wesley (hereinafter *Aho et al.*). For instance, Fig. 1.9 on page 10 of *Aho et al.* shows the phases of such a compilation process, including syntax analysis (or parsing) and code generation. The alternative to compiling is writing the code directly in assembly language or binary machine language, which is typically impractical. Therefore, it would have been obvious to one having ordinary skill in the computer art at the time the invention was made to incorporate parsing and code generation into the disclosed implementation as a widely accepted means to achieve such an implementation.

It is further unclear, based on materials made available to the Examiner, whether or not the cited presentation expressly disclosed generating HTML code that contains a human-readable description of the protocol specification. However, Official Notice is taken that the specific procedures and data packet formats necessary for sending and receiving data for a particular protocol are necessary in order to be able to implement such. One would be motivated to generate documentation of a communication protocol to provide human-readable protocol documentation information to software developers enabling them to implement the protocol. Further, HTML is a platform-independent document format, and one would be motivated to use HTML for the purpose of generating the documentation to allow it to be read on different platforms. Therefore, it would have been obvious to one having ordinary skill in the computer art at the time the invention was made to incorporate the generation of HTML protocol

documentation into the method presented to allow software developers using various computer platforms to read and understand the proper procedures involved in implementing the protocol.

Conclusion

13. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

14. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Eric B. Kiss whose telephone number is (703) 305-7737. The examiner can normally be reached on Tue. - Fri., 7:30 am - 5:00 pm. The examiner can also be reached on alternate Mondays.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Dam can be reached on (703) 305-4552.

Art Unit: 2122

Any response to this action should be mailed to:

Commissioner for Patents
P.O.Box 1450
Alexandria,VA 22313-1450

Or faxed to:

(703) 872-9306 (for formal communications intended for entry)

Or:

(703) 746-7240 (for informal or draft communications, please label
“PROPOSED” or “DRAFT”)

Hand-delivered responses should be brought to Crystal Park II, 2121 Crystal
Drive, Arlington, VA, 22202, Fourth Floor (Receptionist).

Any inquiry of a general nature or relating to the status of this application or proceeding

should be directed to the receptionist whose telephone number is (703) 305-3900.

EBK

September 10, 2003



TUAN DAM
SUPERVISORY PATENT EXAMINER